Google Cloud

A practical guide to data science with Google Cloud

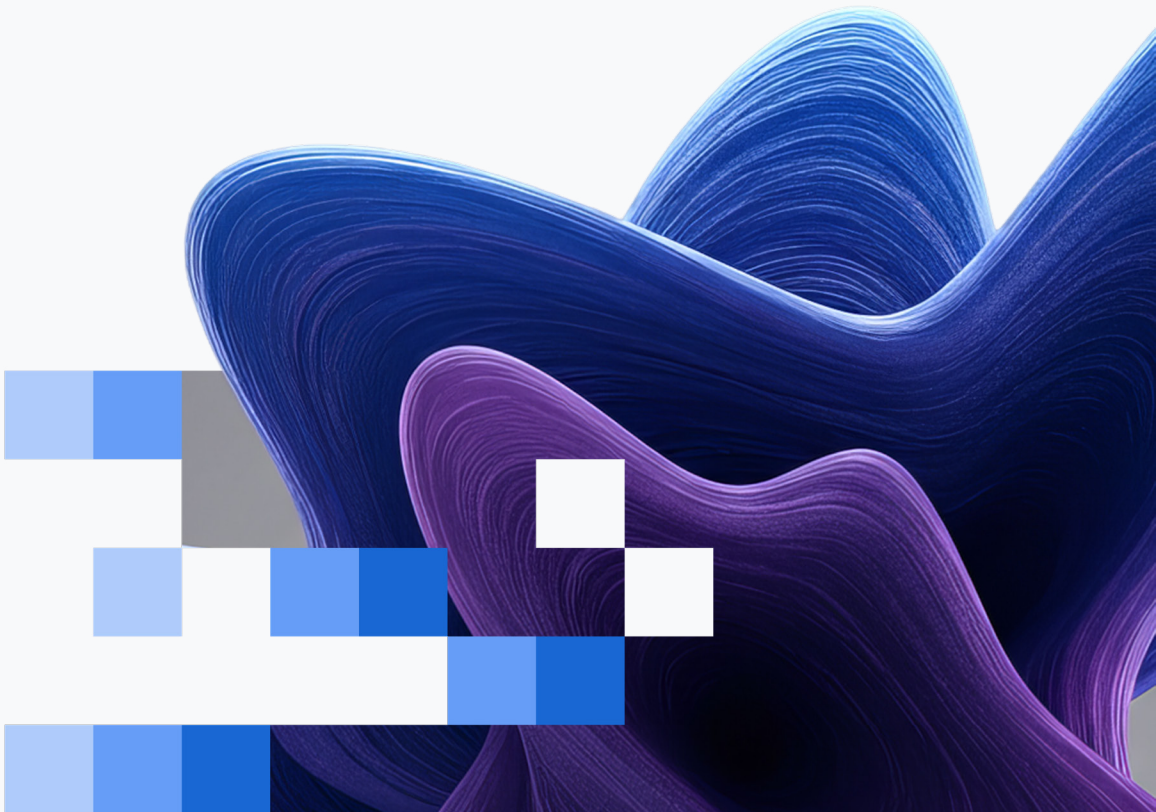


Table of contents

Introduction
The agentic shift, multimodal data, and generative AI 03

Chapter 1
End-to-end data science workflows with Google Cloud 05

Chapter 2
Getting started with data science 14

Chapter 3
A practical guide for data science use cases 16

Introduction

Data science is undergoing its most profound transformation yet, as AI unlocks a wave of new use cases that data scientists can solve faster and more efficiently. With the help of AI, data science teams are now able to automate routine tasks, use previously untapped unstructured data, and achieve new levels of efficiency. This guide helps you get started with data science workflows on Google Cloud and offers a range of real-world use cases (with code) for you to explore.

The agentic shift: Reducing friction and enabling flow

The data science workflow is inherently iterative and often complex, forcing practitioners to constantly switch contexts between creative exploration and time-consuming, repetitive tasks. This mental burden of moving from high-level problem formulation to the granular work of data cleaning, model evaluation, and optimization loops creates friction and breaks analytical flow. AI agents and assistive AI tools help by automating these tedious steps, allowing you to focus your expertise on what matters most: analysis, interpretation, and solving the core business problem.

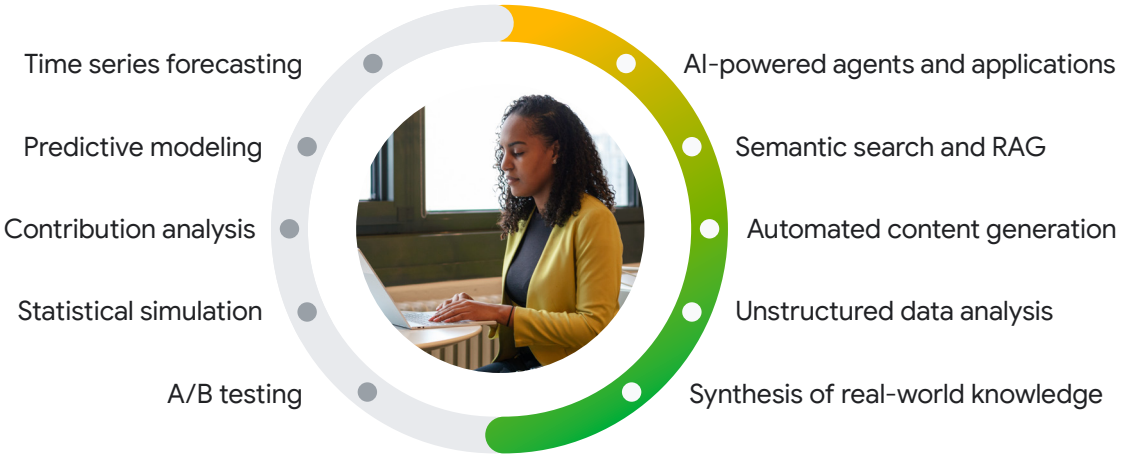
Multimodal data: Unlocking the vast majority of your data

For a long time, valuable insights were locked in unstructured data like images, audio, documents, and text. AI acts as a way to "see" inside images and "understand" the nuances of language at scale. This breakthrough moves us beyond rows and columns, enabling richer questions that combine data types in a single analysis. For example, you can now programmatically score review sentiment or analyze call transcripts to find emerging issues.

Generative AI: Blending real-world knowledge with your enterprise data

AI enriches your analysis by introducing vast real-world knowledge. This is made possible through foundation models, which are pre-trained on a large corpus of general information. These models bring an understanding of concepts and context that augments and enhances your internal data. By blending this external intelligence with your enterprise data, you can uncover deeper insights, and formulate questions that require reasoning. For instance, AI enables you to identify products with high satisfaction "based on quality"—a subjective concept the model can interpret without the need for manually defined rules. This capability allows for more sophisticated and nuanced data analysis.

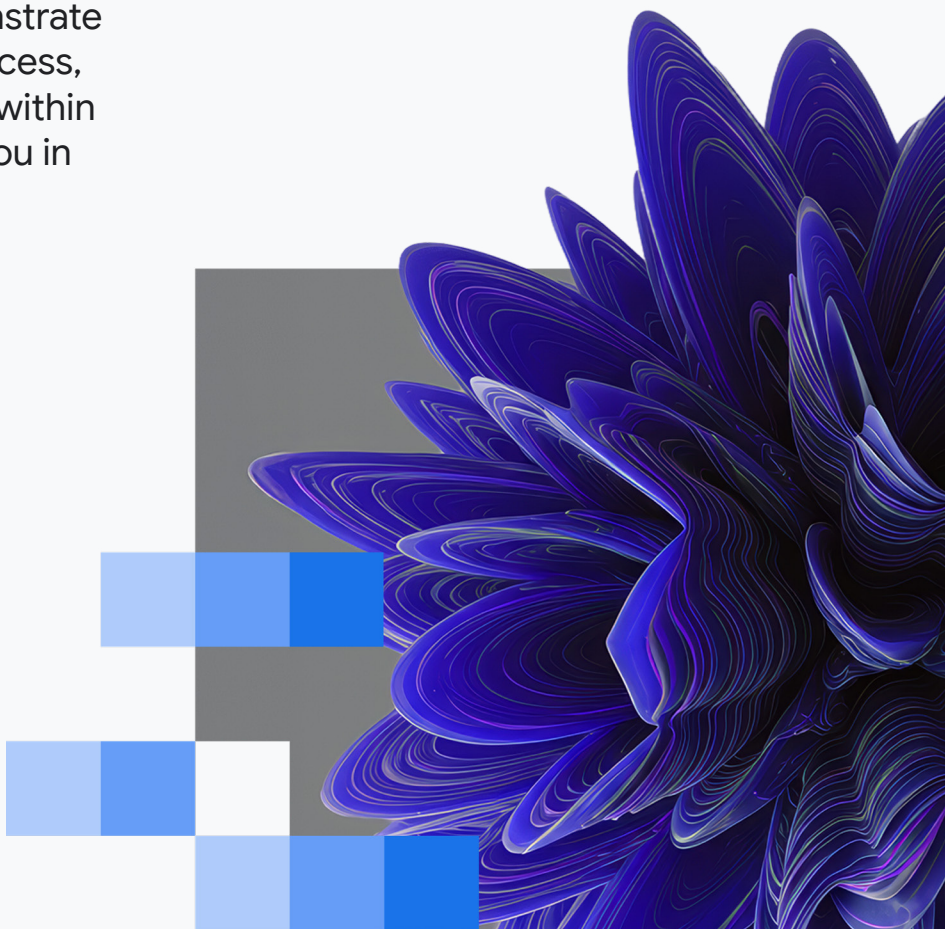
Data scientists can now take on new use cases with the help of AI



Chapter 1

End-to-end data science workflows with Google Cloud

As a data scientist, your goal is to get from a business question to a robust, actionable insight as efficiently as possible. This section provides a practical walkthrough of the integrated Google Cloud tools to support this modern workflow. It will demonstrate how to move from setup to data access, model building, and production, all within a unified ecosystem built to keep you in your analytical flow.



Your central workspace: The AI-first, multilingual notebook

Notebooks are the preferred development environment for data science projects. But a fragmented environment forces constant tool-switching, which disrupts flow. A typical data science workflow might involve writing SQL in one tool for exploratory analysis or data transformation, exporting results, loading that into a local Python notebook, and then using a different tool for visualization.

Google Cloud offers a flexible suite of notebook solutions tailored for enterprise data scientists, accommodating various workflows and governance requirements. For a fully managed and collaborative experience, Colab Enterprise notebooks provide a secure, familiar interface that is deeply integrated and directly accessible within BigQuery and Vertex AI.

For greater customization, Vertex AI Workbench delivers powerful, managed JupyterLab instances that serve as a direct entry point to Serverless for Apache Spark, BigQuery, and Vertex AI. Teams preferring a fully Integrated Development Environment (IDE) experience can use Cloud Workstations with the Cloud Code extension to provision secure, on-demand Visual Studio Code environments. Additionally, Google Cloud supports self-hosted setups, by providing tools like the BigQuery JupyterLab and Visual Studio Code extensions, which enable seamless data exploration and machine learning directly from a user's preferred IDE.

In this section, we'll focus on Colab Enterprise—the preferred notebook experience for BigQuery users.

Seamlessly blend languages

This is where the friction of context-switching disappears. In this polyglot environment, you can query data using SQL in one cell and have the output instantly available as a Python DataFrame in the next. You can also use Spark for large-scale transformations without ever leaving the notebook interface. This eliminates the mental friction of switching between the declarative logic of SQL and the imperative thinking of Python.

An AI-assisted workflow

Colab notebooks are designed with an integrated Data Science Agent to act as an intelligent partner. You can start with a high-level goal in plain English, and the agent can generate a multi-step plan and the corresponding code. You remain in control, reviewing and editing each step.

A practical example

Your prompt

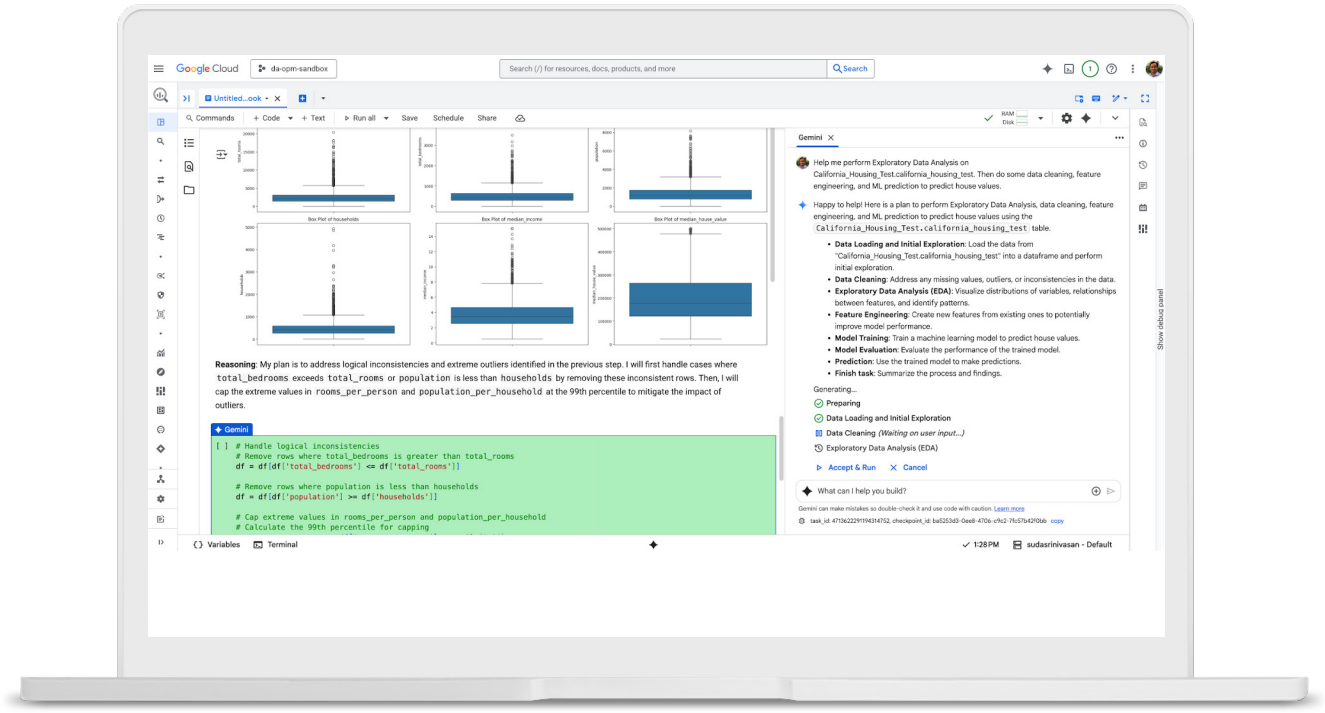
"Train a model to predict 'income_bracket' from table: census_adult_income dataset."

AI-generated plan

1. Load the census_adult_income table from BigQuery
2. Clean the data by removing rows with null values
3. Train a logistic regression model using BigQuery ML to predict income_bracket
4. Evaluate the model's accuracy

The agent then generates the SQL and Python code for each step, which you can execute, modify, and build upon. AI-powered code completion, error explanation and correction, and visualization further simplify the experience.

Colab notebooks offer an AI-powered unified development environment for analysis and rapid experimentation



Accessing and preparing data without silos

Your environment is set up—now you need data. This is where friction often arises, from handling different data types to rewriting code when datasets become too large for a single machine.

A unified, multimodal data foundation

Structured data is typically only a small part of your data estate. Insights are locked in images, documents, and audio files. BigQuery allows you to manage all this data in one place. Using Object Tables and the ObjectRef data type, you can treat unstructured data in Google Cloud Storage as if it were a standard SQL table. This means you can use the power and familiarity of SQL to analyze images or process documents directly. For example, you can call an AI function to extract text from a PDF or analyze an image within your SQL query. All data, whether structured or unstructured, is secured under a single, unified governance model, simplifying compliance.

AI-assisted data preparation

Data scientists spend a lot of time on manual, often repetitive data cleaning and transformation tasks. BigQuery accelerates this process with AI-assisted data preparation. The built-in Gemini-powered editor transforms data preparation from a manual coding exercise into an interactive, AI-guided workflow, allowing you to focus on the logic of your transformations rather than the syntax. As you work with your data, Gemini provides context-aware suggestions for tasks like applying transformations, standardizing data, and automating schema mapping.

The new Data Engineering Agent in BigQuery goes beyond assistive AI and provides autonomous agentic capabilities. The data engineering agent takes instructions in plain English, e.g., “Create a pipeline to load data from the ‘customer_orders’ bucket, standardize the date formats, remove duplicate entries based on order ID, and load it into a BigQuery table named ‘clean_orders’.” Then generates the required SQL code, builds the pipeline, and even creates basic unit tests. The agent can handle a wide range of data transformations and data engineering tasks, including ingestion, data quality, incorporating custom business logic in pipelines, and data modeling.

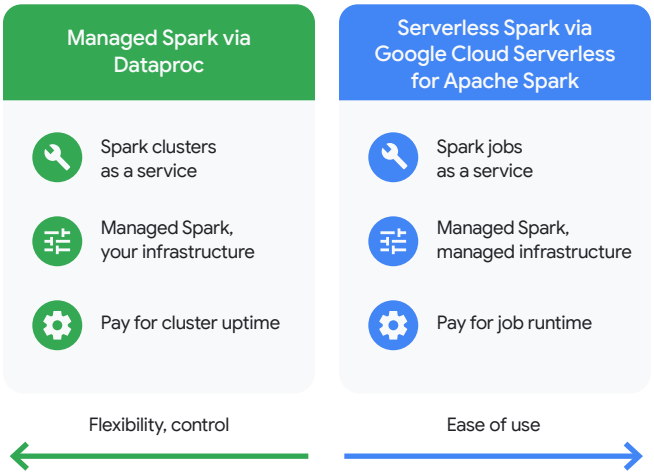
Want to learn more about working with multimodal data in BigQuery?

Explore this notebook with code sample.

Data enrichment and Google third-party datasets

Use the integrated data marketplace to discover and use over 3,500 commercial and public datasets, which can be joined with your own tables without any data movement or ETL processes. Leverage unique public datasets from Google, including the Earth Engine Data Catalog for planetary-scale environmental data, Google Places Insights for location-based intelligence, and Google Trends to analyze search interest over time. Blending your proprietary business data with global-scale external intelligence lets you dig deeper for insights.

Data scientists can use Spark serverlessly or a managed service via Dataproc



Flexible data processing with multiple engines

BigQuery's architecture allows you to bring your preferred processing engine—whether it's BigQuery's SQL engine or an open-source framework like Serverless for Apache Spark—directly to a single, unified copy of your data. This avoids the need to maintain separate data copies for different systems.

You can run serverless Spark directly from your Colab Enterprise or Jupyter notebook. This on-demand environment eliminates the need to provision and manage clusters, offering fast startup and autoscaling. For those who prefer Python-native libraries, BigQuery DataFrames (BigFrames) provide a pandas-like API that translates your Python code into optimized SQL for execution on the BigQuery engine. This gives you the flexibility to use the right tool for the job—whether it's SQL, PySpark, or a pandas-style DataFrame—all while working on the same underlying data.

> “Public Value Technologies receives data feeds from our media partners for our data mesh solution and AI applications. BigQuery data preparation allows our data analysts and scientists to rapidly integrate the data feeds that standardize and preprocess the data in a low code way.”

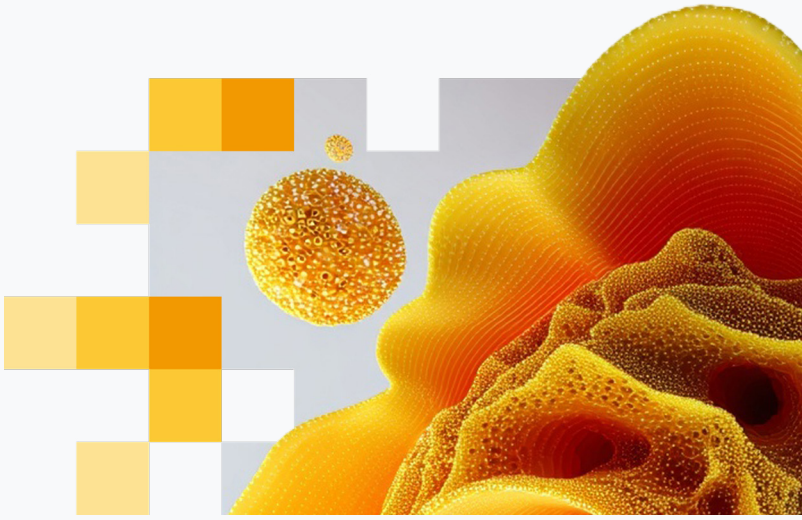
Korbinian Schwinger
Team Lead Data Engineer, Public Value Technologies

A spectrum of model development options

With your data prepared, you can move on to the core task of building and training a model. Google Cloud provides a broad set of tools designed to meet you where you are, whether you prefer the speed of SQL, the power of foundation models, or full control over custom model development.

Bringing AI to your data with BigQuery Machine Learning (BQML)

A significant amount of time in any project is spent engineering pipelines to move data to a separate environment for model training. This creates friction, forcing teams to deal with disparate infrastructure and security policies and disrupting their flow. BQML eliminates this by allowing you to train, evaluate, and deploy machine learning models directly within the data warehouse using familiar SQL commands. This is a practical approach for quickly building models like linear regression, k-means clustering, or time series forecasts without moving data. It also democratizes model building, enabling data analysts who are already SQL experts to develop powerful predictive models.



Train an ML model with SQL

```
CREATE OR REPLACE MODEL `my_dataset.income_prediction_model`
OPTIONS(model_type='LOGISTIC_REG') AS
SELECT
  * EXCEPT(income_bracket),
  income_bracket as label
FROM `bigquery-public-data.ml_datasets.census_adult_income`;
```

> With BigQuery ML, we've accelerated our model deployment to production 10x faster, enabling our sales teams to better serve our Pro customers with prioritized leads and deeper insights."
>
> **Mani Pantangi**
> Senior Manager, Data Science, The Home Depot

BQML offers a wide range of commonly used statistical and predictive machine learning models that are built into BigQuery for ease of use, including contribution analysis, linear regression, classification, k-means clustering, matrix factorization, time series forecasting, and more. These built-in models greatly simplify the most common workflows for data scientists. As an added benefit, you can perform a dry run before creating the model to get an estimate of how much data the model will process to run.

Leveraging pre-trained foundation models

Rather than building every model from scratch, you can "command" powerful, pre-trained foundation models.

Gen AI workflows

BigQuery lets you use SQL functions that call on models like the Gemini family to analyze or enrich your data. For example, you can use the AI.GENERATE_TABLE function to extract structured information from unstructured text. Under the hood, this is a managed remote function call to a Gemini model endpoint, but to you, it's just another SQL function.

Specialized foundation models

For specific tasks, you can use state-of-the-art models. For example, TimesFM, a model from Google Research pre-trained on over 100 billion data points, is integrated into BQML. This lets you generate highly accurate time series forecasts with a simple SQL query, bypassing weeks of custom model development.

Want to learn about using AI functions in BigQuery?

This notebook has code samples to help you get started.

Training models using Spark in BigQuery

If you are familiar with Python and Spark and would like to use your favorite machine learning libraries such as Spark ML, you can train, evaluate and run inferencing on your models directly in BigQuery while leveraging the distributed processing capabilities of Spark.

Custom model development with Vertex AI

For projects that need more control, Vertex AI provides an end-to-end ML platform. Here, you can build, train, and tune custom models using PyTorch, TensorFlow, or any other ML library. The power lies in the seamless integration. You can do feature engineering with BigQuery, train a custom model in Vertex AI, and then use that model for inference directly from BigQuery via SQL.

Vector embeddings and vector search

While not directly related to model training, a common data science task is to find similar items within a large dataset. BigQuery lets you generate and use vector embeddings to perform vector search, enabling semantic understanding and similarity-based retrieval of multimodal data.

The workflow to generate embeddings is straightforward: you create a BigQuery ML remote model that references imported or Vertex AI embedding models. You then use that model with the ML.GENERATE_EMBEDDING function to create embeddings for multimodal data in BigQuery tables. You can create vector indexes using algorithms like inverted file indexing (IVF) or the scalable approximate nearest neighbor (ScaNN) algorithm from DeepMind to accelerate similarity searches. This allows you to build sophisticated semantic search, recommendation, or segmentation systems without needing to manage a separate, specialized vector database. BigQuery is especially useful for these systems because it is all managed serverlessly, regardless of your batch or interactive needs.

To solve inconsistent product matching, Mercado Libre, the Latin American ecommerce leader, used BigQuery Vector Search on product embeddings. It acted as a high-speed filter, rapidly identifying an initial set of similar items from their vast catalog. This crucial first step enabled other AI models to then perform a more detailed analysis on a pre-filtered list.

Quick guide: When to use which tool?

Use BigQuery ML when:

You want to rapidly build and iterate on common model types (regression, classification, clustering, forecasting) directly on your data in BigQuery using SQL. It's ideal for speed and for teams with strong SQL skills.

Use Spark in BigQuery when:

You are familiar with Python and Spark, and are looking to use ML libraries directly for distributed training and inference.

Use Vertex AI when:

You need to build a custom model with a specific architecture, require advanced MLOps capabilities like experiment tracking, or need to use Python frameworks like TensorFlow or PyTorch for deep customization and control.

From model to production with integrated MLOps

A model only provides value when it's operationalized. The “last mile” MLOps challenge of moving a model from a notebook to a scalable, monitored production environment is addressed by a seamless integration between BigQuery and Vertex AI.

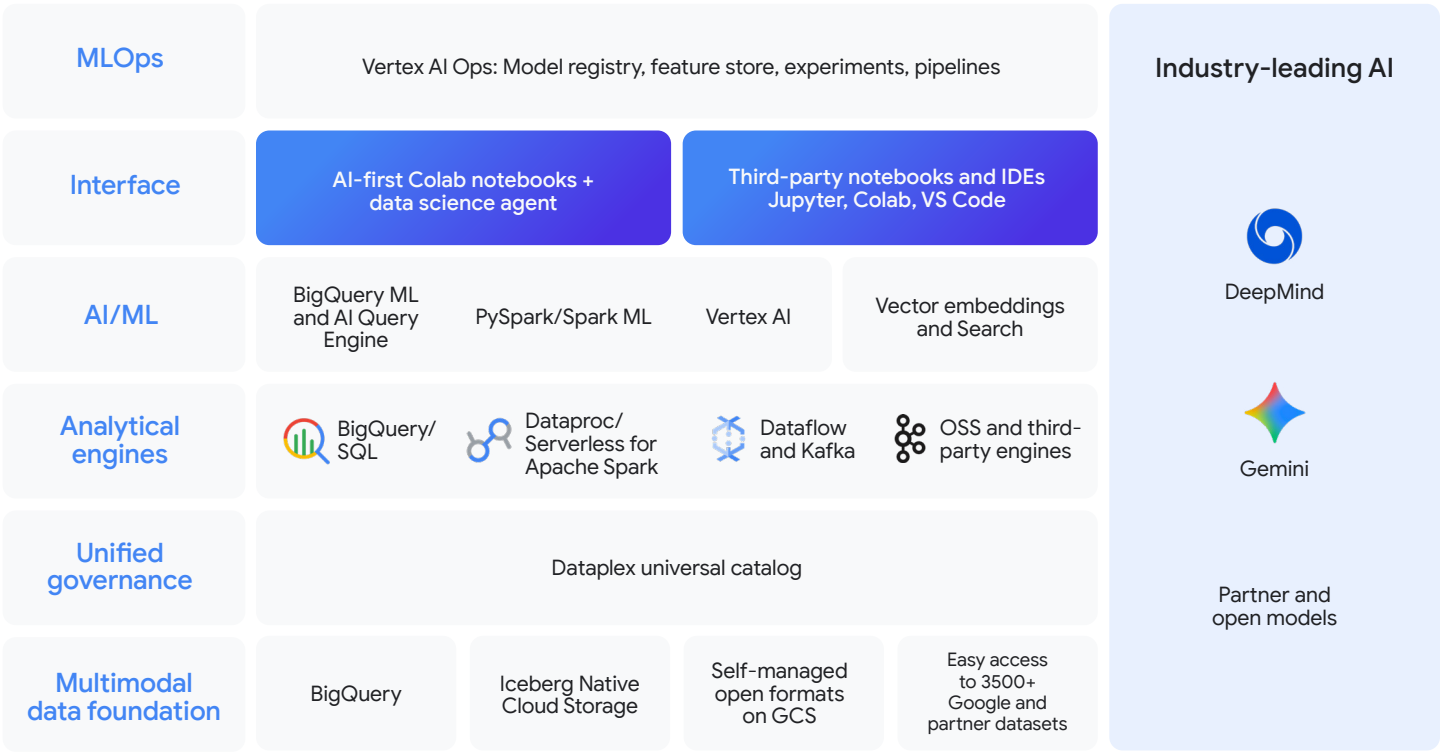
Centralized feature management

You can use BigQuery to conveniently store features for models built and used in BigQuery. For centralized feature management across projects in Google Cloud, you can use the Vertex AI Feature Store to store, share, and reuse features. This prevents redundant data engineering work and, crucially, helps avoid training-serving skew by ensuring that the exact same feature logic is used for both model training and real-time inference. Features can be engineered in BigQuery and then ingested into the Feature Store for consistent access.

Automated model building and management

For tabular data, Vertex AI AutoML automatically handles feature engineering, model selection, and hyperparameter tuning to produce an optimized model. Any model you train, whether in BigQuery ML or Vertex AI, is automatically registered in the Vertex AI Model Registry. From there, with just a few clicks, you can version, evaluate, and deploy your models for batch or online predictions, completing the end-to-end lifecycle from a single, integrated platform.

Google Cloud's data and AI platform for data science and machine learning



Chapter 2

Getting started with data science

Having discussed the key Google Cloud capabilities that streamline and simplify data science workflows, this and the following section will focus on their practical application. We'll first cover the necessary setup to get hands-on and start exploring data science use cases, and then look at a few common examples that data scientists build in Google Cloud.

Google Cloud Account

To start using Google Cloud services, you first need to have an account. If you are new to Google Cloud, you can sign up for a 90-day free trial that includes $300 in free Cloud Billing credits. You can use these credits toward one or a combination of products. Additionally, Google Cloud accounts also provide a free tier to several Google Cloud products including BigQuery and Cloud Storage, allowing you limited access at no cost.

Once you have your Google Cloud Account set up, you will need to decide what notebook platform fits your specific needs. As mentioned in Chapter 1, you have several options to choose from. These include: BigQuery Studio, Vertex AI Workbenches, Cloud Workstations, and self-hosted JupyterLab or Visual Studio Code.

Here's how to get started with each of these options:

BigQuery Studio

To begin using BigQuery Studio:

1. Navigate to the BigQuery page in the Google Cloud console.
2. If prompted, you'll need to enable any required APIs, such as the Dataform API, to activate all of its features.
3. Once the interface loads, in the tab bar of the editor panel, click the down arrow next to the plus sign, and from the floating menu, choose Notebook >> Empty Notebook.
4. If it's your first time creating a code asset, you may be prompted to choose a default storage region.
5. A new tab will then open, presenting you with a blank, untitled Colab notebook, ready for you to start writing Python and SQL code.
6. You also have the option of starting with a template or uploading a notebook from your computer. For more information, please visit the Create notebooks page in our documentation.

Self-hosted JupyterLab

The BigQuery Jupyter extension is a JupyterLab extension that brings powerful Google Cloud features directly into a user's local JupyterLab environment. It allows data scientists to treat BigQuery or Spark on Dataproc as a seamless backend for their notebooks. Key features include the ability to run code on a BigQuery and Spark (serverless or clusters) without any local Spark installation, submit PySpark jobs directly from the notebook, browse and edit objects on Cloud Storage, browse and preview BigQuery datasets, all from within the familiar JupyterLab interface. This dramatically simplifies the workflow for developing and deploying data science pipelines on managed infrastructure.

> **Need to install the BigQuery Jupyterlab plugin?**
>
> Find instructions on the Use the BigQuery JupyterLab plugin page in our documentation.

Self-hosted Visual Studio Code

The Google Cloud Code extension transforms Visual Studio Code into a powerful, integrated development environment for data professionals on Google Cloud. It allows for direct interaction with BigQuery, enabling users to seamlessly browse datasets, inspect schemas, and preview table data without leaving the IDE. For Spark, the extension provides comprehensive Dataproc integration, allowing you to manage the entire lifecycle of your clusters and jobs. You can execute notebooks on Dataproc clusters and serverless, submit jobs to both traditional Dataproc clusters and serverless, and create and manage reusable serverless Spark runtime templates to ensure your notebooks execute with consistent Spark configurations.

> **Ready to install and use?**
>
> Visit the Install the Cloud Code for VS Code extension page in our documentation.
>
> Once you have installed the extension and logged into Google Cloud, you will need to configure three settings:
>
> 1. The project that your notebooks will be executed in
> 2. The BigQuery region
> 3. The Dataproc region that the extension will execute Spark notebooks in

Chapter 3

A practical guide for data science use cases

Now that your data science environment is set up and ready to go, let's look at some practical, scenario-based use cases you can explore using BigQuery and Vertex AI. Each use case starts with a common business challenge and walks through how to use Google Cloud tools to solve it. These use cases explore different statistical and machine learning workflows that data scientists often use.

Use case 1

Get the notebook →

Modernizing retail demand forecasting

The business challenge

In retail, accurate demand forecasting is critical for managing inventory. Overestimating demand leads to high storage costs and wasted product, while underestimating resources results in stockouts and lost sales. The core challenge lies in generating forecasts that are not only accurate, but also timely and highly granular.

A business rarely needs to predict only its total sales. To make effective operational decisions, it must forecast demand for every single item in each specific store or region. Traditionally, this has been a complex and resource-intensive task. It required data science teams to build, manage, and retrain thousands of individual models—often one for each unique product-location combination. This process can be slow, computationally expensive, and difficult to maintain, making it hard for businesses to react quickly to changing market conditions.

This use case uses the public Iowa Liquor Sales dataset. It provides a practical example, containing daily sales records for various products across the state. While it focuses on liquor sales, the principles and techniques are directly transferable to any retail scenario. The fundamental challenge is the same—whether you are forecasting demand for a specific t-shirt size in a particular store, a smartphone model in a city, or a brand of cereal in a specific supermarket. The goal is to find a scalable process that can handle this complexity efficiently, directly where the data resides.

The data science approach

This use case addresses forecasting by implementing and comparing two powerful time series models using BigQuery ML. First, we'll use ARIMA, a well-established statistical model, which requires us to train on historical data before generating a forecast. Then, we use BigQuery's AI.FORECAST function, which uses TimesFM, a Google Research foundation model to produce zero-shot forecasts without a separatetraining step. This approach eliminates the need to move massive datasets to a separate environment for training. By comparing these methods, you can easily evaluate their tradeoffs to determine which one best meets your needs.

High-level flow

- **Prepare the data:** First aggregate raw sales data into two distinct tables—one for a single time series and another for a multiple, granular time series
- **Forecast a single time series:** We walk through forecasting total sales over time using both the ARIMA and TimesFM models
- **Visualize single series results:** The forecasts from both methods are plotted together to assess their performance and differences
- **Forecast multiple time series:** Scale the analysis to forecast sales for each county and item_name combination, demonstrating how to handle thousands of series at once
- **Visualize multiple series results:** Finally, visually compare forecasts for a specific county-item pair to compare the model's performance at a granular level

Use case 2

Get the notebook →

Assessing environmental risks to protect agricultural investments

The business challenge

For agribusinesses and their investors and insurers, accurately quantifying the financial risk of a production shortage is a major challenge. The direct impact of events like a fire or drought is a loss of crop yield, which leads to revenue loss, supply chain disruptions, and financial consequences.

As global natural events become more frequent, operational costs are rising, driven by higher prices and reduced availability of insurance. Current insurance models often rely on broad, regional risk assessments rather than property-specific data. To make informed decisions about capital investment, such as where to buy new land or plant new crops, and to better protect existing assets, stakeholders require a more precise method for evaluating risk.

The business need is a data-driven model that can assess and quantify this risk at a highly granular level, for example, down to an individual parcel of land. Relying on historical data alone is reactive—a forward-looking, dynamic view of risk is essential to guide financial and operational strategy.

The data science approach

This use case leverages the vast repository of public satellite imagery and environmental data available through Google Earth Engine, which is available as rasterized data directly within BigQuery. Analyzing key raster datasets identifies and monitors critical risk factors such as land surface temperature, water availability, and vegetation density (fuel load). This allows for the creation of a dynamic wildfire risk intelligence system. By joining this environmental data with the geographic locations of vineyards and other assets, stakeholders can generate precise risk assessments for specific areas of interest.

High-level flow

- **Access Chilean regions of interest:** Use Overture Maps in BigQuery Sharing to access regional data. Combine this data with official wine production statistics to map regions of interest and their geographical boundaries using the GEOGRAPHY data type in BigQuery.
- **Explore Earth features:** Quickly explore Earth features using Earth Engine's public datasets. Analyze elevation, wildfires, water irrigation, and availability using datasets such as the NASA Digital Elevation Model, GFSAD Cropland, and GLOBathy Global Lakes Bathymetry.
- **Perform raster analysis with ST_RegionStats:** Apply the ST_RegionStats function to compute aggregate statistics (e.g., mean, sum, standard deviation) for raster data within Chilean regions defined by BigQuery geographies. This allows for summarizing raster insights over vector boundaries.
- **Visualize and interpret results:** Export or connect BigQuery results to geospatial visualization tools (e.g., GeoViz, Looker Studio, Looker, Carto) to interpret and communicate findings effectively.
- **Get actionable insights:** Develop a ML model to predict risk areas and take timely actions.

Use case 3

Get the notebook →

Analyzing customer reviews to investigate a product's sales dip

The business challenge

One of the most urgent and ambiguous challenges a business can face is an unexpected dip in sales for a new product. The initial sales dashboards clearly show what is happening—revenue is falling short of forecasts—but they offer no insight into why. The potential causes are numerous and complex. Is it a pricing issue? A competitor's surprise launch? An ineffective marketing campaign? Or is there a fundamental flaw in the product itself?

In these situations, standard data analysis is often not enough. The real story may be hidden in plain sight or buried deep within vast amounts of disconnected data. This is a classic scenario where a data scientist must think like a detective. Their role is not just to report on metrics but to sift through open-ended and often unstructured data, frame specific hypotheses, and then test them to uncover the truth. Time is of the essence—every day of lagging sales puts pressure on the business, and the outcome of the investigation will directly influence critical product and go-to-market decisions. The challenge is to move from a vague business problem to a precise, data-driven explanation that can guide immediate action.

The data science approach

This use case shows how to convert unstructured customer feedback into a powerful, structured signal for predictive modeling. The core of the approach is using gen AI as a sophisticated feature engineering engine. Instead of just performing high-level sentiment analysis, AI directly within BigQuery parses thousands of text reviews and extracts structured, nuanced information—such as sentiment scores for specific product features like "battery life" or "user interface."

This crucial step transforms raw, unstructured text into a queryable, structured format. This new, AI-generated data can then be seamlessly joined with traditional structured business data, such as sales transactions and customer profiles. With this enriched, unified dataset, data scientists can apply familiar and powerful machine learning techniques. By training an XGBoost model, for example, they can move beyond correlation to understand causation, precisely identifying which specific product attributes mentioned in reviews are the most significant drivers of the sales decline. This provides the business with clear, data-driven, and actionable insights to guide product improvements.

High-level flow

- **Explore and visualize data:** Perform descriptive analysis to understand the data and validate that the sales shortfall for V2 of the product is actually true
- **Extract review sentiment using generate AI-powered features:** Use the AI.SCORE function to extract structured sentiment scores for each product feature mentioned in customer reviews
- **Create a unified feature table:** Join the AI-generated sentiment data with product order information and user profiles (e.g., age, gender, urbanicity)
- **Train and evaluate a model:** Train an XGBoost model in BQML to understand how each feature contributes to sales. Evaluate the model's performance using SQL-based evaluation functions
- **Explain the outcome:** Use the model's feature importance scores to explain the poor sales of the latest product version, linking them directly to sentiment in customer reviews

Use case 4

Get the notebook →

Automating contract analysis for risk and compliance

The business challenge

For legal, procurement, and finance departments in large organizations, the sheer volume of legal contracts represents a significant operational bottleneck and a source of risk. Manually reviewing thousands of vendor agreements, sales contracts, and partnership deals to identify risks, track obligations, and ensure compliance is an expensive, slow, and notoriously error-prone process. A single missed renewal date can lead to unwanted service continuation, and an overlooked non-standard liability clause can expose the company to millions of dollars in unexpected risk.

The problem is one of scale and accessibility. Critical information is locked away in unstructured PDF and text documents, making it impossible to perform portfolio-wide analysis. Legal teams cannot easily answer fundamental questions like, “How many of our contracts have a termination for convenience clause?” or “Which agreements are governed by a specific jurisdiction?” without a time-consuming manual review. This reactive approach prevents the business from proactively managing its contractual risks and obligations.

The data science approach

This use case transforms a qualitative, manual task into a quantitative, automated one by leveraging Gemini models directly within BigQuery. By treating unstructured text documents in cloud storage as if they were structured data, we use the power and familiarity of SQL to perform large-scale analysis.

The workflow begins by creating a remote model in BigQuery that points to a Gemini endpoint in Vertex AI. Then use the AI.GENERATE_TABLE function, providing it with a target schema and a natural language prompt. This prompt instructs the model on exactly what information to extract from each contract, such as “Identify the governing law,” “Extract the contract's effective date and termination date,” and “Summarize the limitation of liability clause.”

The model processes each document and populates a new, structured BigQuery table based on the defined schema. This effectively turns a repository of thousands of legal documents into a queryable database. Data scientists and analysts can then use standard SQL to instantly find patterns, identify anomalies, and aggregate risk exposure across the entire contract portfolio. This enables them to build dashboards that monitor key clauses, set up alerts for upcoming renewal dates, and provide leadership with a comprehensive, data-driven view of the company's contractual landscape.

High-level flow

- **Access contract data in BigQuery:** Create a BigQuery external table that directly reads raw contract text files from a Google Cloud Storage bucket.
- **Enable gen AI in BigQuery:** Establish a connection between BigQuery and Vertex AI and create a remote model that points to a Gemini model.
- **Transform text into structured data:** Use AI.GENERATE_TABLE to execute a single SQL query that instructs the Gemini model to parse each document and return structured information.
- **Enrich data with AI summaries:** Use AI.GENERATE to create a concise executive summary for each contract.
- **Analyze and visualize insights:** With the contract data now fully structured and enriched, use standard SQL queries to perform analysis. The results are then visualized using Plotly to create an interactive dashboard that answers key business questions, such as:
  - What is the distribution of different contract types?
  - Which contracts have the highest risk scores?
  - What is the average risk score per contract category?
  - How many contracts fall under specific legal jurisdictions?

Use case 5

Get the notebook →

Predicting changes to the Consumer Price Index

The business challenge

Effective strategic planning requires businesses to understand and anticipate the impact of macroeconomic trends. Among the most critical of these is inflation. The Consumer Price Index (CPI) is a key measure of inflation because it tracks the average change in prices paid by consumers for a basket of goods and services. If the CPI increases, it means that the prices of goods and services have increased on average, signaling a rise in inflation and a corresponding drop in the purchasing power of currency.

This metric is used by a variety of economic actors. Governments use the CPI to set economic policy, such as interest rates. Businesses use the CPI to make critical pricing decisions, manage supply chain costs, and forecast revenue. An unexpected rise in CPI can erode profit margins and alter consumer demand. For this reason, simply having a single-point forecast for the future CPI is often insufficient for robust planning. A single prediction does not capture the inherent uncertainty of the future. The business challenge is to move beyond a single forecast and instead understand a probable range of future scenarios to build more resilient financial and operational strategies.

High-level flow

- **Pre-process historical data:** Filter the historical CPI dataset to prepare it for simulation
- **Run the Monte Carlo simulation:** Use a PySpark job to generate thousands of possible future paths for the CPI based on historical volatility and trends
- **Analyze and plot results:** Aggregate the simulation results and plot the distribution to show the range of possible future CPI paths, highlighting key percentiles (e.g., 20th, 50th, and 80th) to represent different levels of probability

The data science approach

To address the need for a probabilistic forecast, the data science approach is to use simulation, a powerful tool in the data scientist's toolbox for modeling uncertainty. This use case specifically employs the Monte Carlo method to simulate thousands of potential future paths of the CPI. Instead of producing a single number, this technique generates a probability distribution of potential future outcomes.

The Monte Carlo method works by randomly sampling from a probability distribution. In this case, the process begins by calculating the historical mean and standard deviation of the percentage change in the CPI. These statistical parameters are then used to generate a large number of simulated paths for the future CPI. Each path starts with the current CPI value and then evolves by adding a series of normally distributed random increments, with the size of the increments determined by the historical volatility. This technique often uses a model like geometric Brownian motion, a stochastic process used to represent unpredictable movements.

Once the thousands of simulated paths have been generated, they can be aggregated to analyze the probability of various future events. For example, a business can calculate the probability that the inflation rate in a given year will be below a certain target. By analyzing the results, one can visualize the 80th, 50th, and 20th percentiles of the simulated paths, which correspond to optimistic, median, and pessimistic scenarios. This entire computationally intensive simulation can be executed scalably using PySpark directly from a Colab Enterprise notebook, allowing the data scientist to model complex systems without being constrained by the limits of a single machine.

Use case 6

Get the notebook →

Identifying customer segments for targeted marketing

The business challenge

Marketing teams often need to optimize budget allocation to maximize return on investment (ROI). A one-size-fits-all marketing strategy is inefficient, leading to wasted ad spend and low engagement, as the messaging is not relevant to all recipients. The business problem is to develop a systematic, data-driven method for partitioning a customer base into distinct groups based on their behavior, enabling more effective and personalized marketing campaigns.

While basic segmentation using simple demographic data is straightforward, it often fails to capture the more nuanced differences in customer purchasing patterns. The analytical challenge is to move beyond these simple heuristics and identify meaningful segments based on complex behavioral data, such as purchase frequency, monetary value, and product category preferences. The goal is to produce segments that are not only statistically distinct but also interpretable and actionable for the marketing team.

The data science approach

This use case combines unsupervised machine learning with gen AI to create and characterize customer segments. First, apply a k-means clustering algorithm directly within BigQuery ML to efficiently partition the entire customer dataset based on purchasing behavior within BigQuery.

Clustering effectively groups customers and generates a cluster ID. Second, automate the interpretation of these segments to provide business context. Then use a gen AI function to analyze the behavioral data of customers within each cluster and programmatically generate qualitative descriptions, including concise segment name, segment summary and tailored marketing suggestions for each segment.

High-level flow

- **Prepare customer data:** Consolidate and clean customer purchasing data for analysis
- **Train a clustering model:** Create and train a k-means clustering model using a single SQL query in BigQuery ML to group customers into a specified number of segments
- **Assign customers to segments:** Use the ML.PREDICT function to assign each customer to their respective segment
- **Generate segment personas with AI:** Use the AI.GENERATE_TABLE function to analyze the members of each cluster and automatically generate a concise segment name, segment summary and tailored marketing suggestions for each segment

Use case 7

Get the notebook →

Predicting customer purchase intent

The business challenge

For ecommerce retailers, converting website visitors into paying customers is a key goal. These businesses collect vast amounts of behavioral data—from clicks and page views to browsing history and past purchases—to understand and predict which visitors will end up making a purchase. Organizations need to move from a reactive to a proactive engagement model by identifying, in near real-time, which current visitors are demonstrating a high probability of converting.

While simple heuristics, like triggering an offer when a user adds an item to their cart, can be effective, they often fail to capture the more complex and subtle signals of purchase intent. A user's journey is rarely linear, and the true intent may be hidden within sequences of actions across multiple product pages and categories. The analytical challenge is to synthesize high-volume, noisy clickstream data with historical customer information to create a single, reliable predictive signal. The goal is to produce a model that can score a user's intent during their active session, enabling timely and personalized interventions—like offering a chat assistant or a targeted promotion—to encourage conversion. For data science teams, the primary operational challenge is having an environment that allows them to rapidly experiment with features and build these models without being slowed down by complex infrastructure setup or management.

The data science approach

This use case demonstrates how to build a logistic regression classification model using PySpark to predict whether a user will make a purchase. The entire workflow is executed within a Colab Enterprise notebook in BigQuery Studio, taking advantage of the built-in serverless Spark engine. This approach allows our data science team to use familiar PySpark libraries for data exploration and model training directly on data stored in BigQuery, creating a seamless experience from data to model within a single, integrated environment.

High-level flow

- **Load data from BigQuery into Spark:** Use native Spark functionality to load BigQuery data into memory
- **Generate PySpark code:** Gemini generates PySpark code for exploratory data analysis
- **Perform feature engineering:** Using Spark SQL, create the features needed to train the model
- **Train a logistic regression model:** Use PySpark native machine learning library, MLlib, to train a logistic regression model
- **Evaluate and save the model:** Determine validation metrics for the model and save it for future use

Use case 8

Get the notebook →

Creating an image-based home search engine

The business challenge

For ecommerce platforms, particularly those selling products where images factor into the purchase decision, like appliances, furniture or real estate, the limitations of traditional keyword-based search create significant friction in the customer journey. A user often has a clear visual idea of the product they want but may struggle to describe it with a precise set of keywords (e.g., “mid-century modern home with a big front lawn”). This mismatch between visual intent and text-based search can lead to user frustration, poor search results, and ultimately, abandoned shopping carts.

The business objective is to create a more intuitive and effective product discovery experience that aligns with how users think visually. The technical challenge is to build a system that can accept an image as a search query and return a ranked list of the most visually similar products from a catalog of potentially millions of items. Solving this problem of visual search is critical for improving search relevance, increasing customer engagement, and driving higher conversion rates.

The data science approach

This use case implements a large-scale similarity search system using vector embeddings. The problem of visual similarity is treated as one of proximity in a high-dimensional vector space, building the entire workflow on BigQuery's native multimodal capabilities. Start by using a remote foundation model, called from a BigQuery ML SQL query, to convert each product image into a high-dimensional vector embedding. To enable fast querying over millions of vectors, we will create a VECTOR INDEX on the embeddings.

The search functionality is then exposed through the VECTOR_SEARCH function. When a user provides a query image, we will convert it into an embedding and use this function to find the products with the closest embeddings in the indexed catalog, returning a ranked list of visually similar items in real-time.

High-level flow

- **Prepare multimodal product data:** Consolidate structured product information (descriptions, features) and unstructured data (images) into a BigQuery table.
- **Generate image embeddings:** Use a remote model in BigQuery ML to generate high-dimensional vector embeddings for each product image.
- **Create a vector index:** Build a vector index on the embeddings column in your BigQuery product table using the CREATE VECTOR INDEX command to enable fast approximate nearest neighbor (ANN) lookups.
- **Enable semantic search:** Use the VECTOR_SEARCH function in a query. When a user submits an image, it is converted into an embedding, and this function finds the most semantically similar products based on the proximity of their embeddings.
- **Enhance product discovery:** Integrate the vector search results into the ecommerce platform to power “find similar products” features and improve the overall customer search experience.

Ready to take your next steps?

To get started on your data science journey:

- Try BigQuery for free*
- Contact us for expert advice

* Store 10 GB of data and run up to 1 TiB of queries for free per month. New customers also get $300 in free credits to try BigQuery and other Google Cloud products.

Contributors

Suda Srinivasan
Group Product Manager, Google Cloud

Jeff Nelson
Developer Advocate, Google Cloud

Luis Gerardo Baeza
Customer Engineer, Google Cloud

Diogo Kato
Cloud Data Consultant, Professional Services, Google Cloud

Nivedita Kumari
Customer Engineer, Google Cloud

Haroon Dogar
Senior Product Manager, Google Cloud

Brad Miro
Developer Advocate, Google Cloud

Manoj Gunti
Senior Product Marketing Manager, Google Cloud